

# Robust Object Detection with Real-Time Fusion of Multiview Foreground Silhouettes

Ming Xu <sup>1</sup>, Jie Ren <sup>2</sup>, Dongyong Chen <sup>1</sup>, Jeremy S. Smith <sup>2</sup>, Zhechi Liu <sup>2</sup>, Tianyuan Jia <sup>1</sup>

<sup>1</sup> Department of Electrical & Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215123, P. R. China (Email: ming.xu@xjtlu.edu.cn)

<sup>2</sup> Department of Electrical Engineering & Electronics, University of Liverpool, L69 3BX, Liverpool, UK (Email: {j.ren, j.s.smith}@liverpool.ac.uk)

**Abstract**— An object detection algorithm by using multiple cameras is proposed. The information fusion is based on homography mapping of the foreground information from multiple cameras and for multiple parallel planes. Unlike the most recent algorithms which transmit and project foreground bitmaps, it approximates each foreground silhouette with a polygon and projects the polygon vertices only. In addition, an alternative approach to estimating the homographies for multiple parallel planes is presented. It is based on the observed pedestrians and does not resort to vanishing point estimation. The ability of this algorithm to remove cast shadows in moving object detection is also investigated. The results on open video datasets are demonstrated.

**Keywords**— Object detection, data fusion, video signal processing, computer vision

**Corresponding author:** Ming Xu, Email: ming.xu@xjtlu.edu.cn; Phone: +86 512 88161407; FAX: +86 512 88161899.

# 1. INTRODUCTION

Multi-camera video surveillance is receiving more and more attention in the computer vision community. By analyzing information from multiple cameras, it is possible to monitor activities across large or spatially distributed regions such as public transportation systems. Although this approach requires more dedicated cameras, it increases the overall field-of-view, minimizes the effects of dynamic occlusion, enables the localization of targets in 3-D space, and improves the accuracy and robustness of estimation owing to information fusion.

## *1.1 Multi-Camera Information Fusion*

The existing multi-camera surveillance systems can be categorized according to the levels of information fusion for the purpose of detection and tracking. The first category starts tracking with a single camera view and switches to another camera when the system predicts that the current camera will no longer have a good view [1] [2]. As there is limited information exchange between the cameras, such systems have low-level information fusion. The second category of the multiview methods makes measurements, extracts features and/or even tracks targets in each individual camera view; the measurements, features and/or tracks from multiple cameras are then integrated to obtain the global estimates [3–6]. Although these methods attempt to resolve dynamic occlusion, they are still vulnerable to occlusion. The reason is that the measurements and features are extracted from the individual camera views. This premature is vulnerable to occlusion and grouping. These systems are of intermediate-level information fusion. In recent years the third category of multiview methods has emerged, in which the individual cameras no longer extract features but provide foreground bitmap information to the fusion centre. The objects are detected as the visual hull intersections of these foreground

bitmaps from multiple views [7–10]. In [9] homography mapping is used to combine foreground likelihood images from different views to resolve occlusions and determine regions on the ground plane that are occupied by people. The ground plane was later extended to a set of planes parallel to, but at some heights off, the ground plane to reduce false positives and missed detections [11] [12]. Their work achieves good results in moderately crowded scenes. The third category fully utilizes the visual cues from multiple cameras and has high-level information fusion. This paper will focus on the approaches in this category.

### *1.2 The Problems*

#### 1) The Burden in Transmission and Homography Mapping

Although the methods in the third category are robust in coping with occlusion, the costs of mapping foreground images to a reference image are twofold: it brings about a challenging requirement on the bandwidth of multi-camera networks, if the foreground detection and multiview foreground fusion are carried out by different computers; the pixel-wise homographic transformations at image level, for multiple cameras and multiple parallel planes, are very time consuming and dissuades any cheap real-time implementation.

#### 2) Homography Estimation for Multiple Planes

In the recent approaches in the third category, the homography based foreground mapping is induced not only by a single reference plane (e.g. the ground plane) but also by a set of imaginary planes parallel to the reference plane along the normal direction. In [11], the estimation of the multi-plane homographies is based on the vanishing point of the normal direction. The vanishing point was computed by detecting vertical line segments in the scene and finding their intersection in a RANSAC framework. However, in many video surveillance scenarios there are limited vertical line segments which are robustly detectable, sufficiently long and well distributed across the whole images. In addition, it was reported that vanishing

point estimation by parallel line intersection is not precise enough and is very sensitive to small pixel noise [13]. In [12] four vertical poles were placed in the scene, each of which has four landmark points at known heights. The image coordinate of any point along a pole and at a specific height can be calculated from the image projections of the four landmarks. Then the homography for a parallel plane at that height can be estimated from the four points, each of which is on a different pole but at that specific height. This method is restricted in the number of landmark points (the number of poles) for homography estimation and needs pole installation.

### *1.3 The Contributions*

This paper stems from the third-category approaches in that the homography mapping for a set of parallel planes has been used to fuse the foreground information from multiple camera views. The contributions of this paper are as follows:

#### 1). Real-Time Transmission and Homography Mapping of Foregrounds

To accelerate the transmission and projection of the foreground information to a reference image, it is reasonable to focus on foreground regions. However, to warp the foreground regions in a camera view to the reference image, one has to apply the inverse homography to each pixel in the reference image; if it is mapped in a foreground region in that camera view, then it is labeled as a foreground pixel in the reference image. This process is still an image-level homography mapping. As a remedy, we approximate the contour of each foreground region with a polygon. The vertices of the polygon are projected into the reference image through homography mapping. Then the foreground region is rebuilt by filling the polygon projected in the reference image. This greatly saves the network bandwidth and accelerates the processing by avoiding the image-level homographic transformation.

## 2) An Alternative Approach to Estimation of Multi-Plane Homographies

In this paper, the homography estimation for a set of parallel planes at different heights is based on the observed pedestrians. The image coordinates of the feet and the tops of heads of selected pedestrians in each camera view are collected during a training stage. If the cameras are not mounted so high as comparable to their distances to the pedestrians, the image coordinate of any point along the principal axis of a person and at a specific height can be approximated by linear interpolation between those of the feet and the top of head. Then the homography for the parallel plane at that height can be estimated from the interpolated landmarks at that height. This approach is robust in that the number of available landmarks from moving pedestrians is very large. This approach is different from the algorithms in [14] [15], which extract the vanishing point by estimating the intersection of the principal axes of walking pedestrians.

## 3) Cast-Shadow Removal Using Multi-Plane Homographies

Cast shadows due to moving objects have been one of the major challenges in detection and tracking for video surveillance. In this paper, the homography mapping based on multiple parallel planes has been used to detect objects with cast shadows. As the cast shadows are only located on the ground plane, they will not appear as foreground regions in the multi-plane detection results. In contrast, the existing algorithms [16] [17] using the ground-plane homography only cannot discriminate the pedestrians' feet from their cast shadows, because all these regions touch the ground.

The remainder of this paper is organized as follows. In Section 2 the algorithms for the foreground extraction and the polygon approximation in each camera view are introduced. In Section 3 the alternative approach to estimating the homographies for a set of parallel planes is described. In Section 4 the rebuilding and fusion of the projected foreground regions in the reference image are introduced. Section 5 discusses how to use the multi-plane homographic constraints to remove cast shadows in moving-object detection. The experimental results on

open video datasets are given in Section 6, followed by the conclusions.

## 2 FOREGROUND POLYGONS

The foreground detection in each camera view is conducted by using an image differencing operation. To ease the transmission and homography mapping of the foreground information, each foreground region is represented by a polygon which approximates the contour of that region.

### 2.1 Foreground Region Detection

The image differencing operation for foreground detection compares each incoming frame with an adaptive background image and classifies those pixels of significant variation into foregrounds. The probability of observing values  $\mathbf{I}$  at a pixel is modeled by a mixture of Gaussians [18]:

$$P(\mathbf{I}_k) = \sum_i \omega_k^{(i)} G(\mathbf{I}_k, \boldsymbol{\mu}_k^{(i)}, \sigma_k^{(i)}) \quad (1)$$

where  $\boldsymbol{\mu}_k^{(i)}$  is the temporal mean of the  $i$ -th distribution,  $(\sigma_k^{(i)})^2$  is the trace of the covariance matrix, and  $\omega_k^{(i)}$  is the weight reflecting the prior probability that the  $i$ -th distribution accounts for the data. As each pixel process is a non-stationary process and to apply the EM algorithm to each pixel is very time consuming, an on-line K-means approximation is used to update the model. At time  $k$ , every new pixel value is checked against the Gaussian distributions in a mixture model. For a matched distribution, the pixel measurement is incorporated in the estimate of that distribution and the weight is increased:

$$\begin{aligned} \boldsymbol{\mu}_k &= (1 - \rho)\boldsymbol{\mu}_{k-1} + \rho\mathbf{I}_k \\ \sigma_k^2 &= (1 - \rho)\sigma_{k-1}^2 + \rho\|\mathbf{I}_k - \boldsymbol{\mu}_k\|^2 \end{aligned} \quad (2)$$

where  $\rho$  controls the background updating rate and  $\rho \in (0, 1)$ . For unmatched distributions, their estimates remain the same but the weights are decreased. If none of the existing

distributions matches the current pixel value, either a new distribution is created, or the least probable distribution for the background is replaced. The distribution(s),  $i_B$ , with the greatest weight is (are) identified as the *a priori* background model for the next frame. At time  $k$ , the set of foreground pixels identified is:

$$F_k = \left\{ (r, c) : \left\| \mathbf{I}_k(r, c) - \boldsymbol{\mu}_{k-1}^{(i_B)}(r, c) \right\| > 2.5\sigma_{k-1}^{(i_B)}(r, c) \right\} \quad (3)$$

where  $(r, c)$  is the pixel coordinate. The foreground pixel map is then transformed into a foreground region map  $M_k$  by connected component analysis, which is followed by a morphological closing operation to bridge splitting body parts and a size filter to remove false alarms.

## 2.2 Polygon Approximation of Foreground Regions

Once the foreground regions have been identified in a camera view, each foreground region is represented by a polygon which approximates the contour of that region. Suppose the original contour is an ordered set of  $N$  points  $C = \{p_1, p_1, \dots, p_N\}$ . The problem is to find a subset of these contour points that can represent the contour well. The Douglas-Peucker (DP) method [19] has been used for the polygon approximation (see Fig. 1). It starts with the original contour and picks up two extreme points which are the most distant from each other:

$$m, n = \arg \max_{i, j \in [1, N]} \text{dist}(p_i, p_j) \quad (4)$$

These two points are connected with a line, which divides the original contour into two segments. For each of these segments, say segment  $C' = \{p_m, p_{m+1}, \dots, p_n\}$ , it is searched to find the point farthest from the line just drawn. That point is added to the approximation if its distance to the line is over a pre-determined value  $\varepsilon$  that controls the accuracy of the approximation:

$$\begin{aligned}
q &= \arg \max_{i \in [m, n]} \text{dist}(p_i, \overline{p_m p_n}) \\
\text{dist}(p_q, \overline{p_m p_n}) &> \varepsilon
\end{aligned} \tag{5}$$

Then segment  $C'$  is split at point  $p_q$  and the process is recursively applied to the two resultant smaller segments until all the contour points are within distance  $\varepsilon$  to the edges of the polygon. This algorithm can be applied to either convex or concave contours. Moreover, it produces simplification with a hierarchical structure, in which the top layer represents the dominant shape properties and the bottom layer describes the fine details. The most time consuming part of the Douglas-Peucker algorithm is the evaluation of the distances between contour points to line segments. Its worst case running time is  $O(N^2)$  where  $N$  is the number of contour points. An improvement for speeding up the Douglas-Peucker algorithm, making it a  $O(N \log N)$  time algorithm in the worst case, can be found in [20]. Fig. 2 shows some examples of the polygon approximation.

### 3 HOMOGRAPHIC MAPPING

Planar homography is a special relationship defined by a  $3 \times 3$  transformation matrix between a pair of captured images of the same plane with a degree of overlapping:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{6}$$

Let  $(x, y)$  and  $(x', y')$  be a pair of correspondence points on this plane in the two image views.  $\mathbf{x} = [x \ y \ 1]^T$  and  $\mathbf{x}' = [x' \ y' \ 1]^T$  are the homogeneous coordinates. They can be associated with  $\mathbf{H}$ :

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \tag{7}$$

The homography matrix  $\mathbf{H}$  with eight unknowns can be recovered from at least four pairs of corresponding points in the two image views. The more pairs of the corresponding points, the



better estimation  $\mathbf{H}$  obtained. In addition, the estimated homography matrix performs better if these points are homogenously distributed.

When a foreground region in one image view is projected to a top view by homography mapping based on the ground plane, it will be observed as the intersection of the foreground visual hull and the ground plane, like a cast shadow when the camera were replaced with a light source (see Fig. 3). When the foreground regions for the same object are projected to the top view from multiple camera views, the projected foreground regions will intersect in the locations where the object touches the ground plane, e.g. at the feet of the object. The homography mapping based on the ground plane can be extended to a set of imaginary planes parallel to the ground plane and at different heights. For such a plane at the height of a person's waist, the projected foreground regions from multiple image views will intersect at the waist of that person in the top view (see Fig. 3). If such intersection patches by using multiple parallel planes are logically ANDed, the result is similar to the projection of the person's 3D volume on the ground plane.

The estimation of the homographic transformation matrices, from each camera view to the top view, for a set of parallel planes is divided into three steps, as described in subsections 3.1–3.3.

### *3.1 Estimation of the Ground-Plane Homography*

The PETS'2001 dataset [21] was used here, in which the synchronized sequences from two camera views are provided. We used the Google satellite image [22] for the same site as the top-view image and manually selected a set of static landmark pairs on the ground plane in each camera view and the top view. Then the homography matrix  $\mathbf{H}_0$  for the ground plane was estimated.

### 3.2 Homography Estimation for the Top-of-Head Plane

A graphical interface was used to browse the video sequence of each camera view and collect the image coordinates  $\mathbf{x}_f$  for the feet and  $\mathbf{x}_h$  for the top of head of each selected pedestrian (see Fig. 4). Although an automatic tool may be developed by extracting the principal axes of the observed pedestrians, it is not trivial to reliably identify the outliers such as vehicles, pedestrian groups, cyclists, people with a pram or luggage, children, etc. The corresponding image coordinates in the top view can be estimated from those of the feet and the ground-plane homography:

$$\mathbf{x}'_f = \mathbf{H}_0 \mathbf{x}_f \quad (8)$$

By assuming that the selected persons stand upright on the ground and have similar heights, their tops of heads are located on the same plane parallel to the ground plane and at the average height  $h$  of the selected pedestrians. Any minor violation to this assumption can be filtered out in the homography estimation process which finds an optimal solution to fit a large amount of data. Suppose the homography for the top-of-head plane is  $\mathbf{H}_h$ , then the top of head of each selected pedestrian is mapped to the top view image at:

$$\mathbf{x}'_h = \mathbf{H}_h \mathbf{x}_h \quad (9)$$

Due to  $\mathbf{x}'_h \cong \mathbf{x}'_f$  for the selected pedestrians, the homography  $\mathbf{H}_h$  for the plane at the average height of the selected pedestrians can be estimated from a large number of  $(\mathbf{x}'_h, \mathbf{x}_h)$  pairs.

### 3.3 Homography Estimation for Multiple Parallel Planes

If the camera is not mounted so high as comparable to its distance to the targets, the image coordinate of a point along the principal axis of the same person and at a specific height  $h'$  can be approximated by linear interpolation between those of the feet and the top of head:

$$\mathbf{x}_{h'} = ((h - h')/h) \mathbf{x}_f + (h'/h) \mathbf{x}_h, \quad \forall h' \in (0, h) \quad (10)$$

The homography matrix  $\mathbf{H}_{h'}$  for the parallel plane at height  $h'$  can be estimated from a large number of interpolated landmarks  $\mathbf{x}_{h'}$  at that height and the corresponding top-view points  $\mathbf{x}'_{h'}$  by bearing in mind  $\mathbf{x}'_{h'} \cong \mathbf{x}'_f = \mathbf{H}_0 \mathbf{x}_f$ . This approach is robust in that the number of landmark measurements from moving pedestrians is very large. It does not desire many vertical line segments in the scene to estimate vanishing points in the normal direction as in [11]. At the same time, it is not restricted in the small number of available landmark pairs and does not need pole installation as in [12].

Fig. 5 is used to verify the homography estimation. Fig. 5(a) shows the framelets of a small number of foreground regions overlaid on the background image for one of the two camera views and at their original locations. There is no building line segment available in this scenario. Fig. 5(b) is the Google satellite image for the same site and used as the reference image. The feet of the pedestrians in Fig. 5(a) were manually localized and the image coordinates are then mapped to the top view. As the projections of the feet, waist and top of head for the same person coincide in the top view, the back projection of the foot position from the top view to Fig. 5(a) corresponds to a point along the principal axis of that person. If the back projection is based on the homography for the top-of-head plane, it is the top of head in Fig. 5(a). If the back projection is based on the homography for the parallel plane at half the average height of the pedestrians, it is the midpoint of that person in Fig. 5(a). Such calculated tops of heads and midpoints are labeled in Fig. 5(a).

The homography estimation for multiple parallel planes, as described above, is a good approximation when the cameras are not mounted very high. Another algorithm has been developed for homography estimation, which satisfies the cross-ratio invariance in the projective geometry and removes the assumption as above. The first two stages in this algorithm are the same with those in 3.1 and 3.2. The third stage is described in subsection 3.4.

### 3.4 Alternative Homography Estimation for Multiple Parallel Planes

Given the homography estimates  $\mathbf{H}_0$  for the ground plane and  $\mathbf{H}_h$  for the top-of-head plane, for any point  $\mathbf{x}'$  in the top view, we can calculate its corresponding points  $\mathbf{x}_f$  on the ground plane and  $\mathbf{x}_h$  on the top-of-head plane in a camera view. The line connecting  $\mathbf{x}_f$  and  $\mathbf{x}_h$  will point at the vanishing point  $\mathbf{v}$  of the normal direction. Multiple such derived lines corresponding to different locations in the top view will ideally intersect at  $\mathbf{v}$ . The vanishing point  $\mathbf{v}$  can be estimated by minimizing the sum of its squared distances to all these lines. Then the homography induced by a parallel plane is given as in [11]:

$$\mathbf{H}_i = (\mathbf{H}_0 + [\mathbf{0} | \gamma \mathbf{v}]) \left( \mathbf{I}_{3 \times 3} - \frac{1}{1 + \gamma} [\mathbf{0} | \gamma \mathbf{v}] \right) \quad (11)$$

where  $\gamma$  is a scalar multiple proportional to the height of that parallel plane and  $\mathbf{0}$  is a  $3 \times 2$  zero matrix. The homography  $\mathbf{H}_h$  of the top-of-head plane, which is initially estimated in subsection 3.2, will be updated by using (11). Fig. 6(a) illustrates the lines used to estimate the vanishing point in normal direction. The crosses on each line are the intersection points with the ground plane and the top-of-head plane. This result is actually based on subsection 3.2 and on the other hand reflects the accuracy of this approach when compared with the noisy landmarks in Fig. 4. Fig. 6(b) illustrates the accuracy of the multi-plane homography estimation described in this subsection, in the same way as Fig. 5(a).

## 4 FUSION OF FOREGROUND POLYGONS

Once the homography matrices for the set of parallel planes are ready, instead of applying homographic transformations to the foreground images, we only need to project the vertices of the foreground polygons to the reference image. The foreground regions are then rebuilt by filling the internal area of each polygon with a fixed value.

#### 4.1 Filling of Foreground Polygons

In filling the projected polygons, we have to decide whether a given pixel in the top view image lies inside, outside, or on the boundary of a polygon. This is the point-in-polygon problem in computational geometry. In this paper the ray casting algorithm [23] has been used, in which the number of times that a ray (say in horizontal direction) starting from the given point intersects the edges of the polygon is counted (see Fig. 7(a)). If the point in question is not on the boundary of the polygon, it is outside if the number of intersections is an even number; it is inside if this number is odd. However, a vertex of the polygon may fall on the ray or one side of the polygon may lie entirely on the ray (see Fig. 7(b)). To avoid duplicate counts of the edge crossing, if the intersection point is a vertex of a polygon side being tested, then the intersection is counted only if the second vertex of the side lies below the ray. The time to test one point against a polygon with  $L$  sides or  $L+1$  vertices is  $O(L)$ . This algorithm can be applied to either convex or concave polygons.

#### 4.2 Fusion of Foreground Regions

Suppose that the foreground region map for camera view  $c$  is  $M_c$  and the homographic transformation matrix, from camera view  $c$  to the top view  $T$ , for parallel plane  $p$  is  $\mathbf{H}_p^{c,T}$ , then the rebuilt foreground region map, projected from camera view  $c$  according to the homography for plane  $p$ , is denoted by:

$$M_T^{c,p} = \mathbf{H}_p^{c,T}(M_c) \quad (12)$$

For a specific plane  $p$ , the fusion of the foregrounds in the top view is carried out by overlaying the foreground region maps from all the camera views:

$$M_T^p = \sum_c M_T^{c,p} \quad (13)$$

The highlights in  $M_T^p$  correspond to the intersection patches of the moving objects with plane  $p$  and are denoted by:

$$I_T^p = \bigcap_c M_T^{c,p} \quad (14)$$

For the ground plane, the intersection patches are in locations where the moving objects touch the ground. Fusion of the foreground information can be further carried out by overlaying the foreground region maps projected to the top view according to the homographies for all the parallel planes:

$$M_T = \sum_p M_T^p = \sum_p \sum_c M_T^{c,p} \quad (15)$$

The highlights in  $M_T$  are like the projection of the 3D volumes of the moving objects on the ground plane and are denoted by:

$$I_T = \bigcap_p I_T^p = \bigcap_p \bigcap_c M_T^{c,p} \quad (16)$$

In the implementation as above, the objects in the overlapping field of views (FOVs) will be favoured, because they receive foreground votes from multiple cameras. The objects visible in only a single camera view may be lost, if a global threshold is applied to the foreground fusion image. As an alternative solution, the pixel values can be doubled or the threshold can be halved within the regions which are visible in only one of the two camera views. Suppose that the FOVs projected from the two cameras to the top view are represented by binary masks  $F_1$  and  $F_2$  respectively, the FOV visible to only a single camera is the pixel-wise exclusive-OR of  $F_1$  and  $F_2$ :

$$F = F_1 \oplus F_2 \quad (17)$$

## 5 A CASE STUDY IN CAST-SHADOW REMOVAL

Cast shadows due to moving objects have been one of the major challenges in detection and tracking for video surveillance. They are often misclassified as foregrounds, which distort the object shapes and cause adjacent objects to “merge” with each other. This brings difficulties

to tracking, because the observations for the individual objects in a group of merged objects cannot be readily extracted [24].

There exist some algorithms in detecting shadows from image sequences. The photometric approach has been widely used for shadow detection, which assumes that cast shadows reduce luminance values while maintaining chromaticity values of the background pixels. However, it is found that part of real foreground regions may satisfy this definition and be missed in the detection. In addition, the cast shadows in outdoor scenes are bluish, due to the scattered light by the sky, rather than maintain the chromaticity values of the background [25]. Good surveys in monoview cast shadow detection and removal can be found in [26] [27].

Multiple cameras have been employed to remove or detect cast shadows. Although this approach requires more dedicated cameras, it improves the accuracy and robustness of the detection owing to information fusion. Onoguchi [16] proposed a method by using two camera views and assuming that moving objects are standing on the ground plane. Then one camera view is warped to the other by a homographic transformation based on the ground plane. The pixel values in one camera view and the warped image from the other camera view are compared. If the pixel values from these two images are highly correlated, then the underlying pixel is determined as the background or a background appearance change such as a cast shadow. In [17] Lanza et al extracted the change mask image in each of multiple view images by using a background subtraction algorithm. The change mask images are projected to a virtual top view image by homographic transformations. The intersections of these projected change masks from multiple views correspond to the ground plane locations of people as well as their cast shadows. Then the intersection regions are warped back to and subtracted from the single-view change masks. However, these two approaches remove not only the shadows but also the pedestrians' feet, because all these regions touch the ground.

To solve this problem, the homography mapping induced by multiple parallel planes is

used to detect objects with cast shadows. As the cast shadows are only located on the ground plane, they will not intersect any parallel plane off the ground and thus disappear in the detection using multi-plane homographies. In contrast, the torso of a pedestrian will intersect all these parallel planes. Therefore, the feet can be discriminated from the cast shadow of the same person.

The intersection patches for the ground plane and those for the multiple planes are warped back to the single camera views, according to the ground-plane homography:

$$M_c^{T,0} = (\mathbf{H}_{c,T}^0)^{-1}(I_T^0) \quad (18)$$

$$M_c^T = (\mathbf{H}_{c,T}^0)^{-1}(I_T) \quad (19)$$

The former is subtracted from the single-view change masks by a set difference operation so as to remove cast shadows:

$$D_c = M_c - M_c^{T,0} \quad (20)$$

However, the feet of the objects are also lost in this process. This is compensated by adding the back-warped multi-plane intersection patches. These intersection patches are dilated by a square structure element  $B$  beforehand, because they reflect the narrowest sections of the moving objects in the logical AND operation:

$$F_c = D_c \cup (M_c^T \oplus B) \quad (21)$$

## 6 RESULTS

The new algorithm has been tested over a range of video sequences which contain significant dynamic occlusion and scene activity. Both qualitative and quantitative evaluations have been carried out by using the PETS'2001 dataset, in which the original sequences were spatially sub-sampled to half-PAL (384×288 pixels). The top view image is of 500×500 pixels.



### 6.1 Performance Evaluations

We have compared the polygon projection and the bitmap projection in the results and processing speeds. Fig. 8 shows some examples of the polygon projection and the bitmap projection, in which the pre-determined distance  $\varepsilon$  for polygon approximation was set to 1 pixel. It is found that they are very close to each other. For more accurate results, this distance  $\varepsilon$  can be set to sub-pixels.

In testing the processing speeds, we run the polygon projection and the bitmap projection on a single PC with Intel Core 2 Duo CPUs of 2.66 GHz. Both the implementations include (1) the foreground detection in two camera views and (2) the projection and fusion of foreground information from the two camera views. Then the time spent for processing each frame from one camera view was obtained by taking the average (see Table 1). Usually in a video surveillance network part 1 is executed by individual clients and part 2 is executed by a central server. Part 1 is not related to the improvement in the new algorithm. It was implemented using either the running average algorithm or the Gaussian Mixture Model algorithm. The running average algorithm takes 15.6 ms and the Gaussian Mixture Model takes 65.0 ms to process one frame for one camera. Therefore, the former is more appropriate for real-time applications. Part 2 was implemented using either the bitmap projection method or the polygon projection method. The polygon projection method is further divided into four stages: polygon approximation, vertex projection, polygon filling and foreground addition to the top view image. Since our implementations in [28], great efforts have been made to optimize the code and accelerate the bitmap projection method. The bitmap projection takes 108.5 ms and the polygon projection takes 8.5 ms to process one frame for one camera. Therefore, the latter is 12.8 times faster than the former.

Foreground Detection	Running Average (ms)	Gaussian Mixture Model (ms)	
	15.6	65.0	
Foreground Projection and Fusion	Bitmap Projection (ms)	Polygon Projection (ms)	
	108.5	Polygon Approximation	4.5
		Vertex Projection	0.1
		Polygon Filling	2.3
		Foreground Addition	1.6
		Sub-Total	8.5

Table 1: The times for running different algorithms for one camera.

Although the bitmap projection method seems not slow, it still dissuade any cheap real-time implementation. The computational burden in fusing foreground visual hulls lies in the homography mapping for multiple cameras and multiple parallel planes. The more cameras and more planes, the more accurate and more robust for the object localization. As an example, four camera views and ten parallel planes were used in [11]. For an implementation with moderate use of resources, suppose two camera views and four parallel planes are being used. Then the bitmap projection will take 868 ms (1.15 fps) and the polygon projection will take 68 ms (14.7 fps) to process one frame. Therefore, it is a great boost in computational speeds. To further accelerate the polygon projection, the algorithm for speeding up the Douglas-Peucker algorithm in [20] can be used.

## 6.2 Experimental Results in Dynamic Occlusion

Fig. 9 illustrates the results of the algorithm in the case of dynamic occlusion. The original images from the two camera views of Dataset 1 are in Figs. 9(a) and 9(c), which are overlaid with the foreground polygons. The polygons are represented in green, while the vertices are in

red. The detected foreground regions in the two camera views are in Figs. 9(b) and 9(d). Fig. 9(e) is the fusion of the foreground polygons in the top view by using the ground-plane homography. The grey regions represent foreground regions observed by a single camera, while the black regions are those observed by both camera views and correspond to the feet of the pedestrian or the bottom of the vehicle. The rectangular region on the left is the projection of the vehicle on the top of camera view 2.

The homography mapping for four parallel planes has been applied to the same sequences. The four planes are at 0% (the ground plane), 25%, 50% and 75% of the average height of the pedestrians, respectively. Fig. 9(f) is the fusion of the foreground polygons in the top view by using the four-plane homographies, in which the darkest regions represent the locations of the pedestrian and the vehicle. Fig. 9(g) is the overlay of the detection result (in red) on a synthetic top-view image, which was generated by warping and fusing the two camera views. Although the pedestrian is occluded by the vehicle in one camera view, they are well separated by fusion of the foreground regions. For the vehicle on the left of the top-view image, it is within the regions visible to camera view 2 only and thus a halved threshold is applied. The detected foreground region clearly corresponds to the bottom of that vehicle.

### *6.3 Experimental Results in Cast Shadows*

Fig. 10 illustrates the results of the algorithm in the presence of cast shadows. The original images from the two camera views of Dataset 3 are in Figs. 10(a) and 10(d), which are overlaid with the foreground polygons. The foreground regions in the two camera views are in Figs. 10(b) and 10(e). Fig. 10(g) is the fusion of the foreground polygons in the top view by using the ground-plane homography. The grey regions represent foregrounds observed by a single camera, while the black regions are those observed by both camera views, corresponding to the feet and cast shadows of the pedestrians. Fig. 10(g) results from the scheme in which the regions covered

by both camera views are favoured; otherwise, the whole foreground region of the pedestrian who is visible only in camera view 2 will be thought as touching the ground. The detection results by using the ground-plane homography are warped back to Figs. 10(b) and 10(e), which are shown in black. The feet and the cast shadows are identified together as the location of the pedestrians, which leads to inaccurate object localization. The pedestrian visible to a single camera is lost in the detection.

The homography mapping for four parallel planes has been applied to the same sequences. The four planes are at 0% (the ground plane), 25%, 50% and 75% of the average height of the pedestrians, respectively. Fig. 10(h) is the fusion of the foreground polygons in the top view by using the four-plane homography mapping. Fig. 10(i) is the synthetic top view overlaid with the detection result in red. The intersection regions are relatively big, which is caused by the similar viewing angles of the two cameras. Figs. 10(h) and 10(i) result from the scheme in which the regions visible to a single camera have the pixel values doubled. The pedestrian on the bottom-right corner is correctly detected. The detection results by using the multi-plane homographies are warped back to Figs. 10(c) and 10(f), which are shown in black. Only the feet of the pedestrians are detected and the pedestrian visible to only one camera view is also detected.

Fig. 11 illustrates the results of applying the algorithm to the *campus* sequences [29]. Only two of the three camera views were used. The homography mapping is based on five parallel planes: the ground plane and planes at 10%, 20%, 30%, 40% of the average height of the pedestrians. These planes are at relatively lower heights, because the cameras were mounted at the average height of the pedestrians. Fig. 11(a) is a virtual top view by warping and fusing the two camera views. The original images are shown in Figs. 11(b) and 11(c). The single-view change masks are shown in Figs. 11(d) and 11(e). They are projected and intersect in the top view with the ground-plane homography, as shown in Fig. 11(f). Warping the intersection

patches back to the single views leads to Figs. 11(g) and 11(h). The change masks by removing ground-plane appearance changes are shown in Figs. 11(i) and 11(j), in which the feet are lost. The intersection of change masks with multi-plane homographies is shown in Fig. 11(k). Warping the multi-plane intersection patches back to the single views leads to Figs. 11(l) and 11(m). The final foregrounds are in Figs. 11(n) and 11(o), in which the cast shadows disappear but the feet remain.

## 7 CONCLUSIONS

We have proposed an efficient object detection algorithm by using multiple cameras. This work is based on multi-plane homography mapping of the foreground polygons from multiple cameras. The experimental results have shown that this algorithm can run in real time and generate results similar to those by mapping foreground bitmaps. In addition, we have proposed an approach to estimating the homographies induced by multiple planes parallel to the ground plane. This method is based on the pedestrians in the video sequences. This algorithm can effectively eliminate cast shadows from moving object detection.

## ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of China (NSFC) under Grant 60975082 and the Natural Science Foundation of Jiangsu Province, China, under Grant BK2008180.

## REFERENCES

- [1] Q. Cai and J. Aggarwal, “Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams,” Proc. IEEE ICCV, 1998.
- [2] S. Khan and M. Shah, “Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View”, IEEE Trans. PAMI, 25(10), 1355-1360, 2003.

- [3] J. Kang, I. Cohen, and G. Medioni, "Continuous Tracking within and across Camera Streams", Proc. IEEE CVPR, 2003.
- [4] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal Axis-Based Correspondence between Multiple Cameras for People Tracking," IEEE Trans. PAMI, 29(4), 663-671, 2006.
- [5] J. Black and T. Ellis, "Multi-camera image tracking", Image and Vision Computing, 24(11), 1256-1267, 2006.
- [6] M. Xu, J. Orwell, L. Lowey, and D. Thirde, "Architecture and Algorithms for Tracking Football Players with Multiple Cameras", IEE Proc. Vision, Image and Signal Processing, 152(2), 232-241, 2005.
- [7] A. Mittal and S. Larry, "M<sup>2</sup>tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," International Journal of Computer Vision, 51(3), 189-203, 2003.
- [8] J. Franco and E. Boyer, "Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid," Proc. ICCV, 2005.
- [9] S.M. Khan and M. Shah, "A Multi-View Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint", Proc. ECCV, 2006.
- [10] J. Berclaz, F. Fleuret, and P. Fua, "Robust People Tracking with Global Trajectory Optimization," Proc. IEEE CVPR, 2006.
- [11] S.M. Khan and M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes", IEEE Trans. PAMI, 31(3), 505-519, 2009.
- [12] R. Eshel and Y. Moses, "Tracking in a Dense Crowd using multiple cameras", International Journal of Computer Vision, 88(1), 129-143, 2010.
- [13] R. Cipolla, D. P. Robertson, E. G. Boyer, "Photobuilder-3 Models of Architectural Scenes from Uncalibrated Images", Proc. IEEE Int. Conf. Multimedia Computing and Systems, 1999.
- [14] F. Lv, T. Zhao and R. Nevatia, "Self-calibration of a camera from video of a walking human", Proc. ICPR, 2002.
- [15] W. Ge and R. T. Collines, "Crowd detection with a multiview sampler", Proc. IEEE CVPR,

2009.

- [16] K. Onoguchi. "Shadow elimination method for moving object detection". Proc. ICPR, 1998.
- [17] A. Lanza, L. Stefano, J. Berclaz, F. Fleuret and P. Hua, "Robust multi-view change detection", Proc. BMVC, 2007.
- [18] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", Proc. IEEE CVPR, 1998.
- [19] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required for represent a digitized line or its caricature", Canadian Cartographer, 10(2), 112-122, 1973.
- [20] J. Hershberger and J. Snoeyink, "Speeding up the Douglas-Peucker line-simplification algorithm", In Proc. Symp on Data Handling, pages 134-143, 1992.
- [21] <http://www.cvg.rdg.ac.uk/datasets/index.html>
- [22] <http://maps.google.com/>
- [23] I. Sutherland, R. F. Sproull and R. Schumacker, "A characterization of ten hidden-surface algorithms", ACM Computing Surveys, 6(1), 1-55, 1974.
- [24] M. Xu, T. Ellis, S. J. Godsill and G. A. Jones, "Visual tracking of partially observable targets with suboptimal filtering", IET Computer Vision, 5(1), 1-13, 2011.
- [25] N. Martel-Brisson and A. Zaccarin, "Moving cast shadow detection from a Gaussian mixture shadow model," Proc. IEEE CVPR, 2005.
- [26] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video", IEEE Trans. PAMI, 26(8), 1079-1087, 2004.
- [27] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithm and evaluation," IEEE Trans. PAMI, 25(7), 918-923, 2003.
- [28] M. Xu, J. Ren, D. Chen, J. Smith and G. Wang, "Real-time detection via homography mapping of foreground polygons from multiple cameras", Proc. IEEE ICIP, 2011.
- [29] <http://cvlab.epfl.ch/data/pom/>

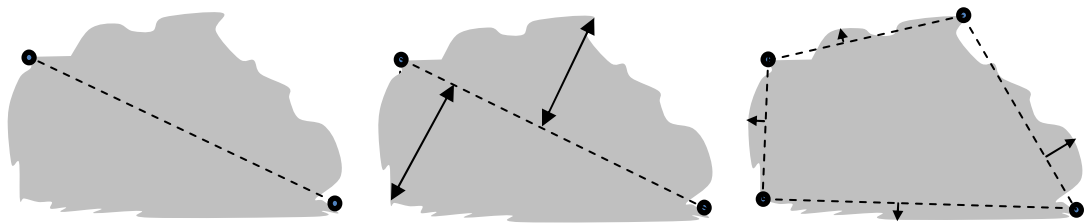


Figure 1: The polygon approximation for a foreground region.



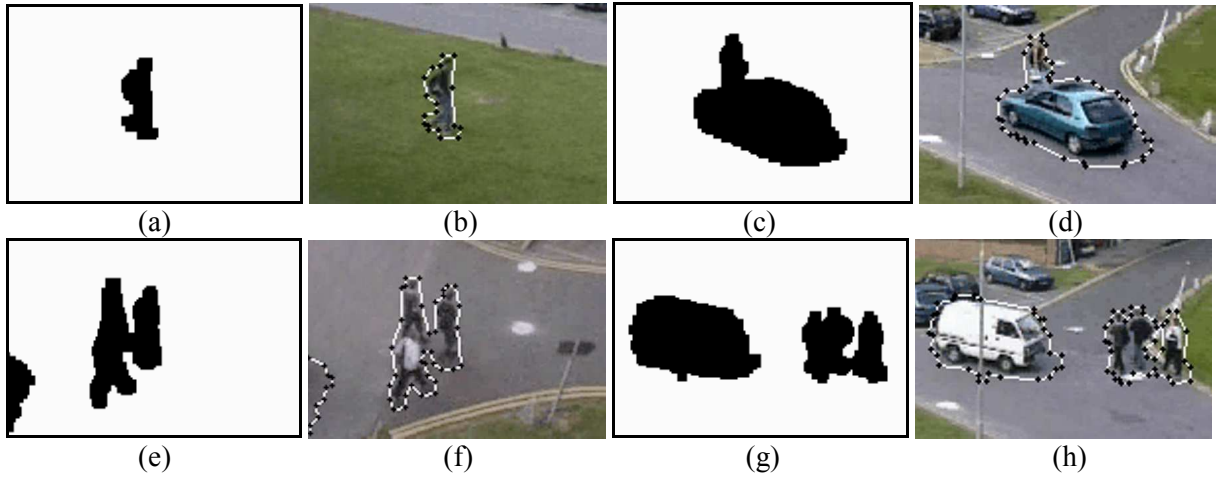


Figure 2: The foreground regions detected in the individual camera views (left) and the corresponding foreground polygons (right). The black points are the vertices.

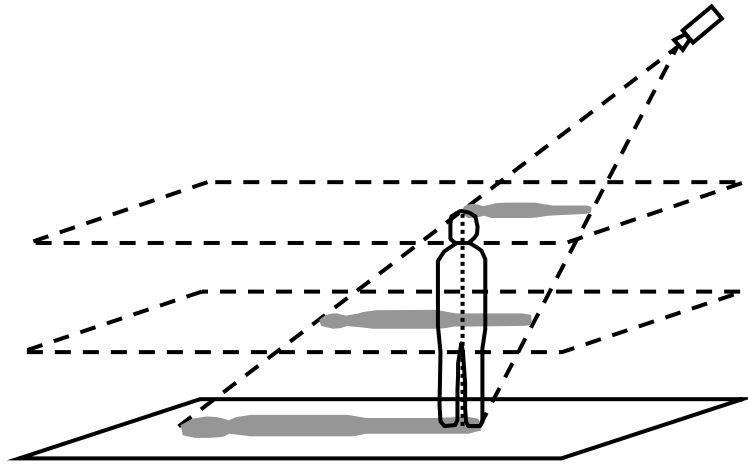


Figure 3: Homography mapping with multiple planes.



Figure 4. A GUI interface to collect the image coordinates of the feet and tops-of-heads of selected pedestrians in video sequences.



Figure 5. The verification of homography estimation for multiple parallel planes: (a) an individual camera view and (b) the top view from Google maps. The feet of the pedestrians in (a) are manually localized and projected to (b) through the ground-plane homography. Then they are back projected to (a) through homography induced by planar planes parallel to the ground plane.

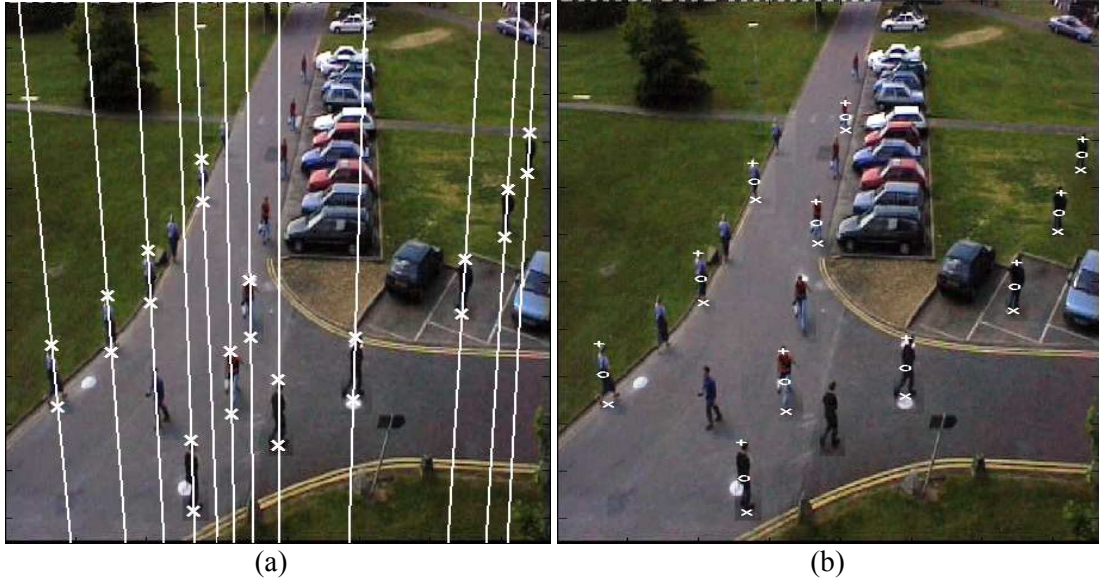
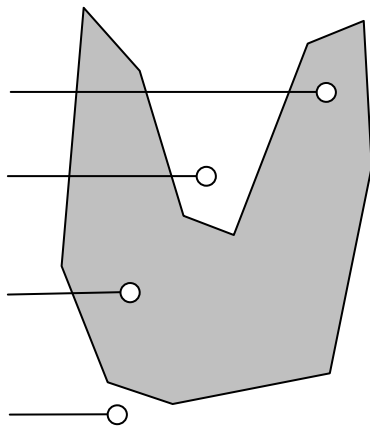
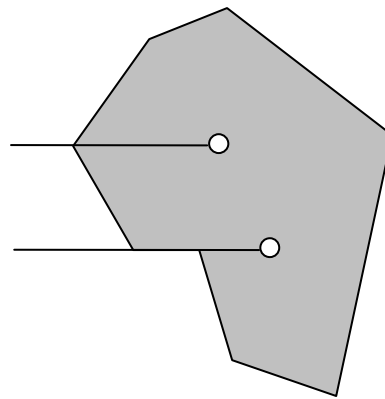


Figure 6. (a) The imaginary lines used to calculate the vanishing point in vertical direction.  
 (b) The verification of multi-plane homography estimation in the same way as in Fig. 4.



(a)



(b)

Figure 7: The ray casting algorithm to decide whether a given point is inside a polygon: (a) when the ray crosses the edges, and (b) when the ray crosses a vertex or lie on an edge.

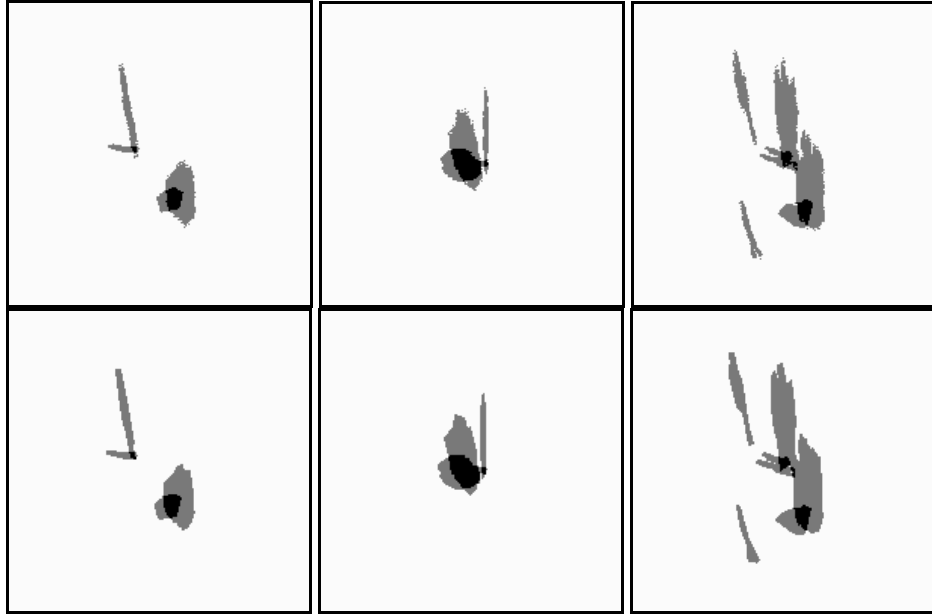


Figure 8: The top-view foreground regions by using bitmap projection (top) and by using polygon projection (bottom).

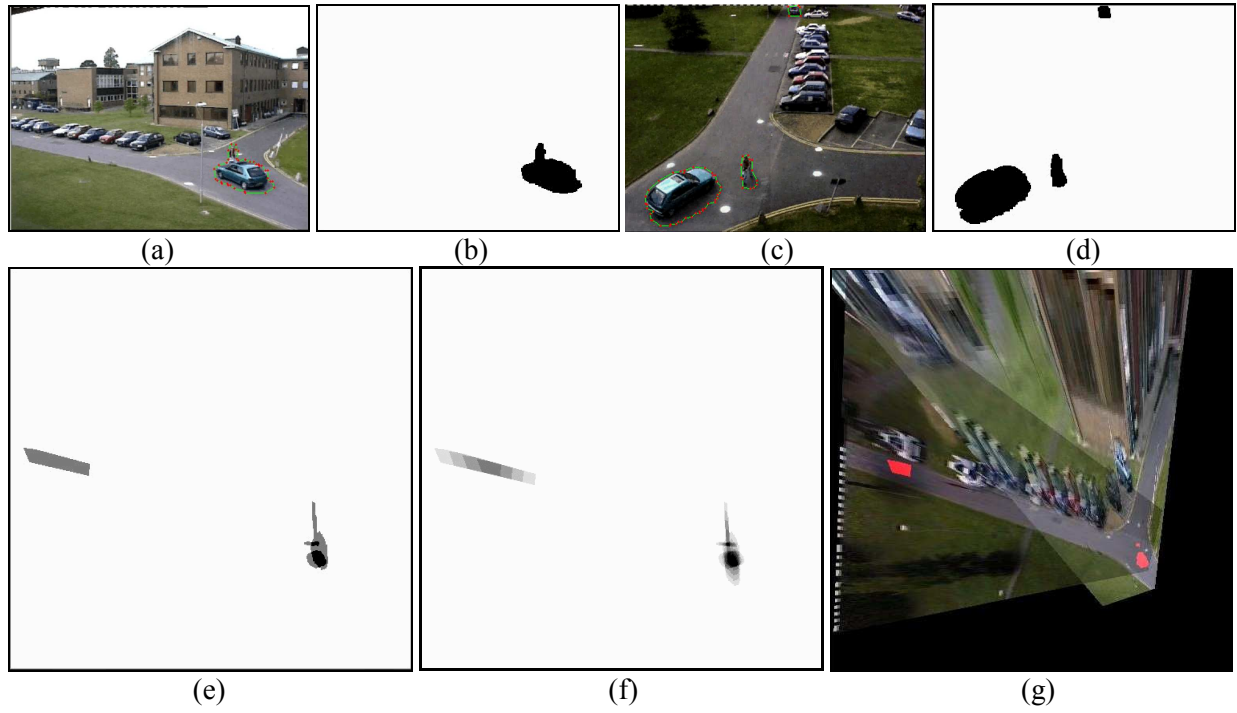


Figure 9: (a)(c) The original images in two camera views with foreground polygons overlaid, (b)(d) the foreground regions in two camera views, (e) fusion of the foreground polygons in the top view using ground-plane homography, (f) fusion of the foreground polygons using multi-plane homography, and (g) the synthetic top view overlaid with the detection results from (f).



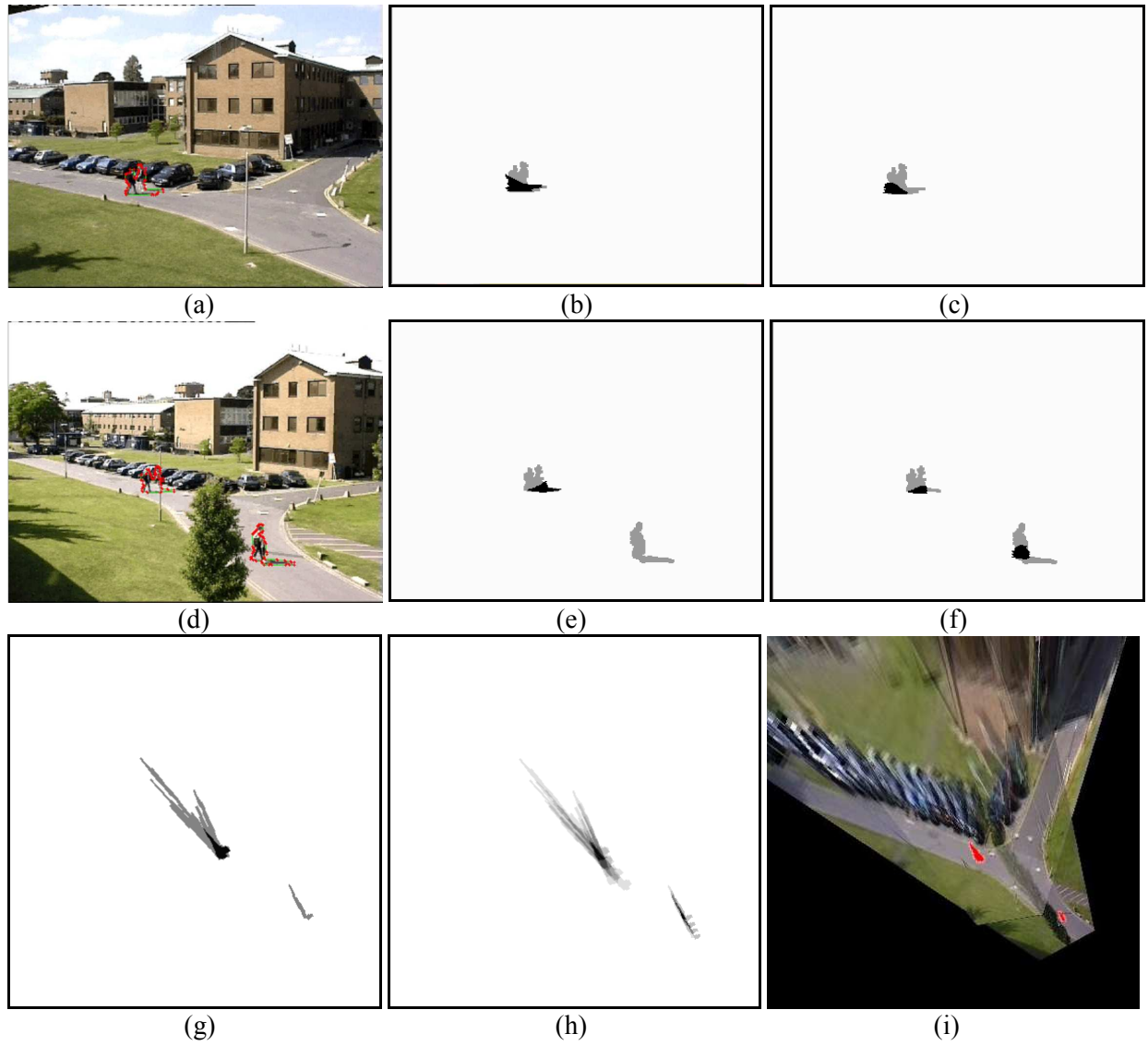


Figure 10. (a)(d) The two camera views with foreground polygons overlaid, (b)(e) the foreground regions overlaid with the warped detection results (in black) by using ground-plane homography, (c)(f) the foreground regions overlaid with the warped detection results (in black) by using multi-plane homography, (g) fusion of the foreground polygons by using ground-plane homography, (h) fusion of the foreground polygons by using multi-plane homography, and (i) the synthetic top view overlaid with the detection results from (h).

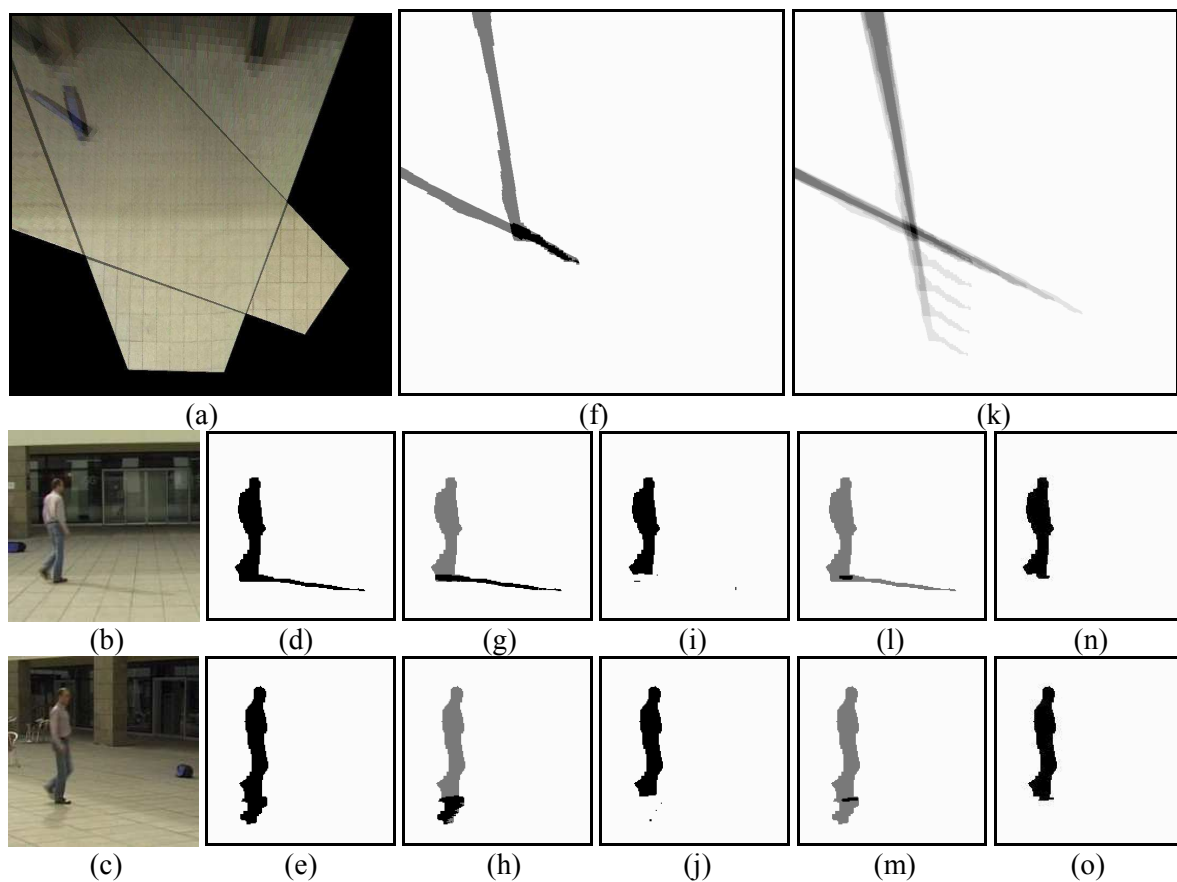


Figure 11: Cast shadow removal with the three rows corresponding to a virtual top view and the two camera views: (a) the virtual top view, (b)(c) original images, (d)(e) single-view change masks, (f) change mask intersection using ground-plane homographies, (g)(h) intersections in (f) warped back to single views, (i)(j) change masks with ground-plane appearance changes removed, (k) change mask intersection using multiple-plane homographies, (l)(m) intersections in (k) warped back to single views, and (n)(o) final foregrounds.